Getting Started with Interface Builder

Session 409

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Getting Started with Interface Builder

Joey Hagedorn
Interface Builder Engineer

View
(Latitude, Longitude)
North America
Atlantic Ocean
Europe
Africa
South America
Google

- (id)init {
if (self = [super init]) {
element = nil;
atomicElementView = nil;
atomicElementFlippedView = nil;
self.frontViewIsVisible=YES;
self.hidesBottomBarWhenPushed = YES;
}
return self;
}
- (void)loadView {
// create and store a container view
UIView *localContainerView = [[UIView alloc] initWithFrame:
[[UIScreen mainScreen] applicationFrame]];
self.containerView = localContainerView;
[localContainerView release];
containerView.backgroundColor = [UIColor blackColor];
}
.m

View
(Latitude, Longitude)
North America
Atlantic Ocean
Europe
Africa
South America
Google

Demo

Creating a user interface

Recap

Creating a user interface

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

```objc
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);

    self.navigationBar = [[UINavigationBar alloc] init];
    self.navigationBar.barStyle = UIBarStyleBlack;
    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);

    UIToolbar *toolbar = [[UIToolbar alloc] init];
    [toolbar setFrame:navigationRectArea];

    UINavigationItem *topItem = [[UINavigationItem alloc] initWithTitle:@"(Latitude, Longitude)"];
    [self.navigationBar pushNavigationItem:topItem animated:NO];

    UIBarButtonItem *sanFranciscoButton = [[UIBarButtonItem alloc] init];
    sanFranciscoButton.style = UIBarButtonItemStylePlain;
    sanFranciscoButton.image = [UIImage imageNamed:@"GoldenGate.png"];
    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    UIBarButtonItem *worldButton = [[UIBarButtonItem alloc] init];
    worldButton.style = UIBarButtonItemStylePlain;
    worldButton.image = [UIImage imageNamed:@"Globe.png"];
    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    UIBarButtonItem *flexibleSpace = [[UIBarButtonItem alloc]
                    initWithBarButtonSystemItem:UIBarButtonSystemItemFlexibleSpace target:nil action:nil];
    NSArray *toolbarItems = [NSArray arrayWithObjects:worldButton, flexibleSpace, sanFranciscoButton, nil];

    toolbar = [[UIToolbar alloc] init];
    toolbar.barStyle = UIBarStyleBlack;
    [toolbar setItems:toolbarItems animated:NO];
    [toolbar sizeToFit];

    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);
    self.map = [[MKMapView alloc] initWithFrame:mapFrame];
    [self.map setMapType:MKMapTypeHybrid];
    [self.map setDelegate:self];

    [self.view addSubview:self.navigationBar];
    [self.view addSubview:map];
    [self.view addSubview:toolbar];

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Creating and Configuring UI

```objc
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);

    self.navigationBar = [[UINavigationBar alloc] init];
    self.navigationBar.barStyle = UIBarStyleBlack;
    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);

    UIToolbar *toolbar = [[UIToolbar alloc] init];
    [toolbar setFrame:navigationRectArea];

    UINavigationItem *topItem = [[UINavigationItem alloc] initWithTitle:@"(Latitude, Longitude)"];
    [self.navigationBar pushNavigationItem:topItem animated:NO];

    UIBarButtonItem *sanFranciscoButton = [[UIBarButtonItem alloc] init];
    sanFranciscoButton.style = UIBarButtonItemStylePlain;
    sanFranciscoButton.image = [UIImage imageNamed:@"GoldenGate.png"];
    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    UIBarButtonItem *worldButton = [[UIBarButtonItem alloc] init];
    worldButton.style = UIBarButtonItemStylePlain;
    worldButton.image = [UIImage imageNamed:@"Globe.png"];
    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    UIBarButtonItem *flexibleSpace = [[UIBarButtonItem alloc]
                    initWithBarButtonSystemItem:UIBarButtonSystemItemFlexibleSpace target:nil action:nil];
    NSArray *toolbarItems = [NSArray arrayWithObjects:worldButton, flexibleSpace, sanFranciscoButton, nil];

    toolbar = [[UIToolbar alloc] init];
    toolbar.barStyle = UIBarStyleBlack;
    [toolbar setItems:toolbarItems animated:NO];
    [toolbar sizeToFit];

    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);
    self.map = [[MKMapView alloc] initWithFrame:mapFrame];
    [self.map setMapType:MKMapTypeHybrid];
    [self.map setDelegate:self];

    [self.view addSubview:self.navigationBar];
    [self.view addSubview:map];
    [self.view addSubview:toolbar];

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Creating and Configuring UI

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);

    self.navigationBar = [[UINavigationBar alloc] init];
    self.navigationBar.barStyle = UIBarStyleBlack;
    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);

    UIToolbar *toolbar = [[UIToolbar alloc] init];
    [toolbar setFrame:navigationRectArea];

    UINavigationItem *topItem = [[UINavigationItem alloc] initWithTitle:@"(Latitude, Longitude)"];
    [self.navigationBar pushNavigationItem:topItem animated:NO];

    UIBarButtonItem *sanFranciscoButton = [[UIBarButtonItem alloc] init];
    sanFranciscoButton.style = UIBarButtonItemStylePlain;
    sanFranciscoButton.image = [UIImage imageNamed:@"GoldenGate.png"];
    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    UIBarButtonItem *worldButton = [[UIBarButtonItem alloc] init];
    worldButton.style = UIBarButtonItemStylePlain;
    worldButton.image = [UIImage imageNamed:@"Globe.png"];
    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    UIBarButtonItem *flexibleSpace = [[UIBarButtonItem alloc]
                    initWithBarButtonSystemItem:UIBarButtonSystemItemFlexibleSpace target:nil action:nil];
    NSArray *toolbarItems = [NSArray arrayWithObjects:worldButton, flexibleSpace, sanFranciscoButton, nil];

    toolbar = [[UIToolbar alloc] init];
    toolbar.barStyle = UIBarStyleBlack;
    [toolbar setItems:toolbarItems animated:NO];
    [toolbar sizeToFit];

    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);
    self.map = [[MKMapView alloc] initWithFrame:mapFrame];
    [self.map setMapType:MKMapTypeHybrid];
    [self.map setDelegate:self];

    [self.view addSubview:self.navigationBar];
    [self.view addSubview:map];
    [self.view addSubview:toolbar];

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);



    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);


    [toolbar setFrame:navigationRectArea];







    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);




    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);



    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);









    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);

    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);

    [toolbar setFrame:navigationRectArea];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Placement and Sizing

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);

    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);

    [toolbar setFrame:navigationRectArea];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Placement and Sizing

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    CGRect viewBounds = [self.view bounds];
    CGFloat viewHeight = CGRectGetHeight(viewBounds);
    CGFloat viewWidth = CGRectGetWidth(viewBounds);

    [self.navigationBar sizeToFit];
    CGFloat navigationBarHeight = self.navigationBar.frame.size.height;
    CGRect navigationRectArea = CGRectMake(0, 0, viewWidth, navigationBarHeight);

    [toolbar setFrame:navigationRectArea];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    CGFloat toolbarHeight = toolbar.frame.size.height;
    CGFloat toolbarYCoord = viewHeight - toolbarHeight;
    CGRect toolbarRectArea = CGRectMake(0, toolbarYCoord, viewWidth, toolbarHeight);
    [toolbar setFrame:toolbarRectArea];

    CGFloat mapFrameWidth = viewWidth;
    CGFloat mapFrameHeight = viewHeight - navigationBarHeight - toolbarHeight;
    CGRect mapFrame = CGRectMake(0, navigationBarHeight, mapFrameWidth, mapFrameHeight);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];
```

```
    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);
```

```
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

```objc
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

Getting Started with Interface Builder

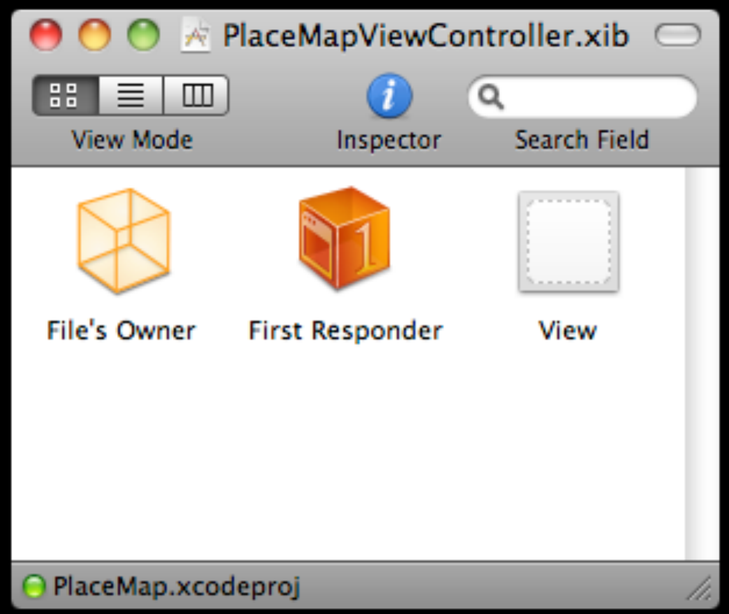
Kelly Keenan
Interface Builder Engineer

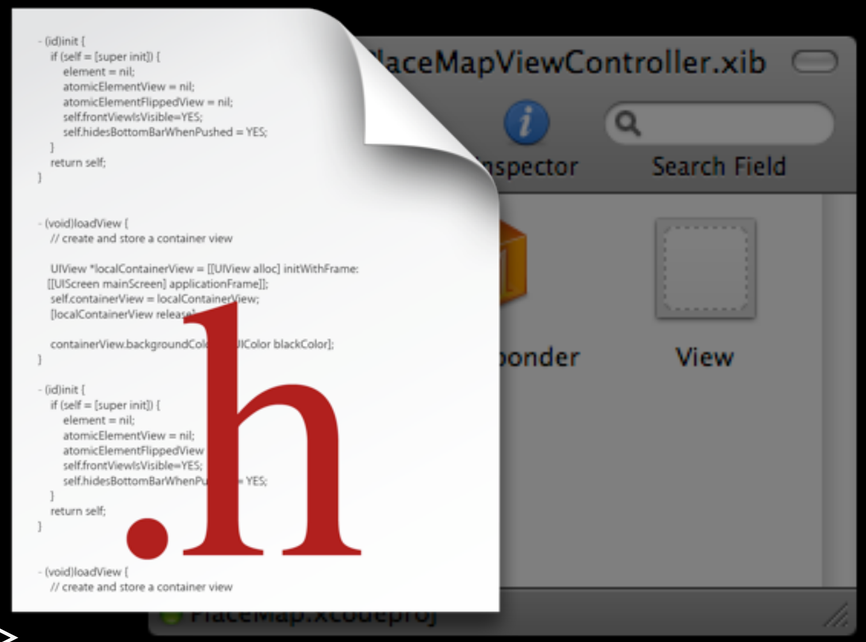
View
(Latitude, Longitude)
North America
Europe
Atlantic Ocean
Africa
South America
Google

- (id)init {
if (self = [super init]) {
element = nil;
atomicElementView = nil;
atomicElementFlippedView = nil;
self.frontViewIsVisible=YES;
self.hidesBottomBarWhenPushed = YES;
}
return self;
}
- (void)loadView {
// create and store a container view
UIView *localContainerView = [[UIView alloc] initWithFrame:
[[UIScreen mainScreen] applicationFrame]];
self.containerView = localContainerView;
[localContainerView release];
containerView.backgroundColor = [UIColor blackColor];
}
.m

Interface Builder and Xcode

Integrating with Xcode





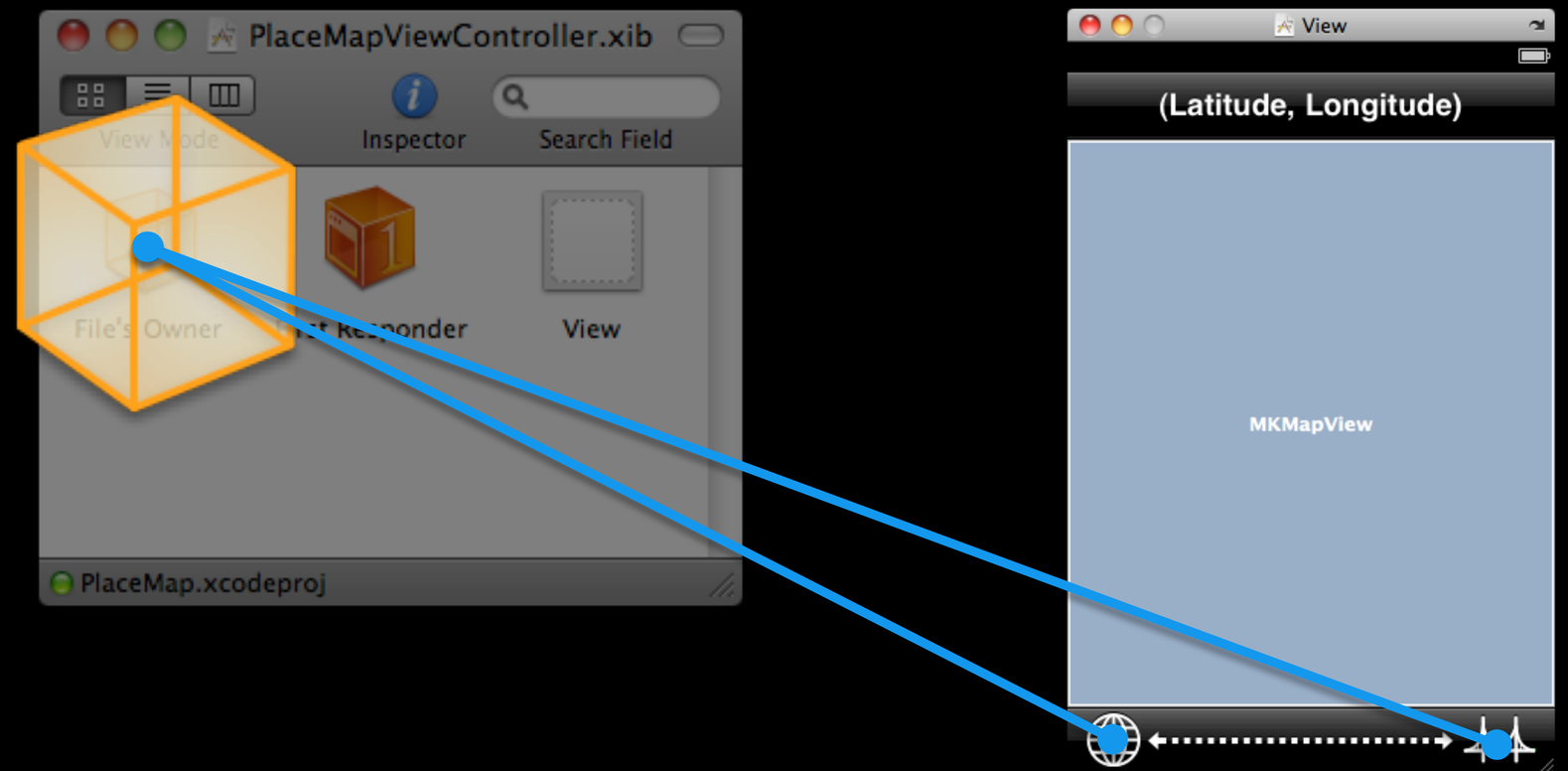

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) IBOutlet MKMapView *map;
@property (retain, nonatomic) IBOutlet UINavigationBar *navigationBar;

- (IBAction)zoomIn:(id)sender;
- (IBAction)zoomOut:(id)sender;

@end
```



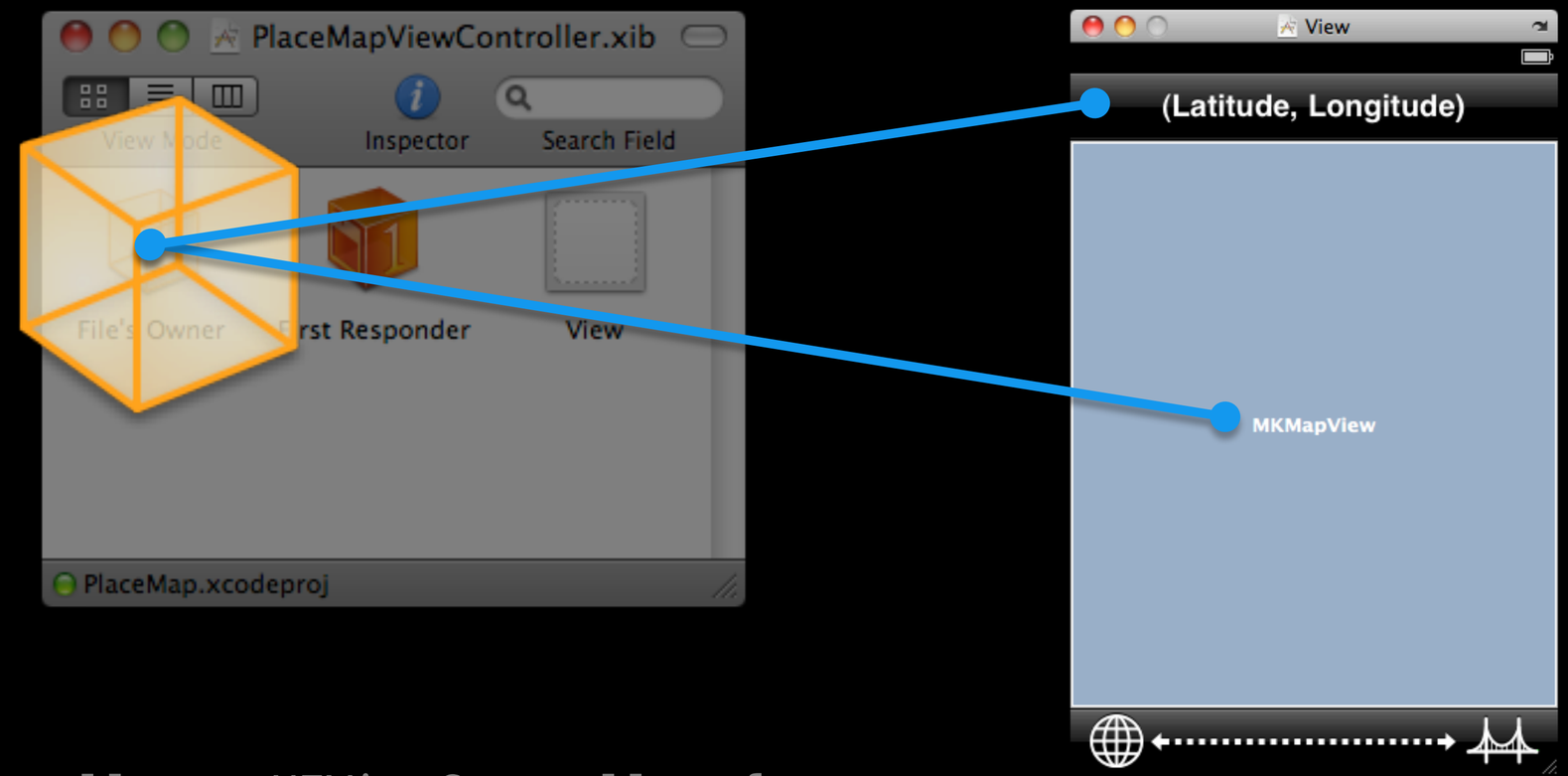

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) IBOutlet MKMapView *map;
@property (retain, nonatomic) IBOutlet UINavigationBar *navigationBar;

- (IBAction)zoomIn:(id)sender;
- (IBAction)zoomOut:(id)sender;

@end
```




```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) IBOutlet MKMapView *map;
@property (retain, nonatomic) IBOutlet UINavigationBar *navigationBar;

- (IBAction)zoomIn:(id)sender;
- (IBAction)zoomOut:(id)sender;

@end
```

Memory Management

```
- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) IBOutlet MKMapView *map;
@property (retain, nonatomic) IBOutlet UINavigationBar *navigationBar;

- (IBAction)zoomIn:(id)sender;
- (IBAction)zoomOut:(id)sender;

@end
```

Demo

Making connections

Recap

Making connections

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Setting Targets and Actions

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Setting Targets and Actions

```
- (void)loadView {
    [super loadView];

    [self setView:[[UIView alloc] initWithFrame:[[UIScreen mainScreen] applicationFrame]]];

    sanFranciscoButton.target = self;
    sanFranciscoButton.action = @selector(zoomIn:);

    worldButton.target = self;
    worldButton.action = @selector(zoomOut:);

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];

    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Creating Window and Controller

```
- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]] autorelease];
    self.mainViewController = [[PlaceMapViewController alloc] init];
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}
```

Creating Window and Controller

```objc
- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {


    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

```objc
#import <UIKit/UIKit.h>
#import "PlaceMapViewController.h"

@interface PlaceMapAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    PlaceMapViewController *mainViewController;
}

@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) PlaceMapViewController *mainViewController;

@end
```

```objc
#import "PlaceMapAppDelegate.h"

@implementation PlaceMapAppDelegate

@synthesize window;
@synthesize mainViewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application {
    [self.window addSubview:[self.mainViewController view]];
    [self.window makeKeyAndVisible];
}

- (void)dealloc {
    [mainViewController release];
    [window release];
    [super dealloc];
}

@end
```

```objc
#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>

@interface PlaceMapViewController : UIViewController <MKMapViewDelegate> {
    MKMapView *map;
    UINavigationBar *navigationBar;
}

@property (retain, nonatomic) MKMapView *map;
@property (retain, nonatomic) UINavigationBar *navigationBar;

- (void)zoomIn:(id)sender;
- (void)zoomOut:(id)sender;

- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;

@end
```

```objc
#import "PlaceMapViewController.h"

@implementation PlaceMapViewController

@synthesize map;
@synthesize navigationBar;

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)updateLocationString:(MKCoordinateRegion)region {
    NSString *locationString = [NSString stringWithFormat:@"(%f, %f)", region.center.latitude, region.center.longitude];
    [[self.navigationBar topItem] setTitle:locationString];
}

- (void)dealloc {
    [map release];
    [navigationBar release];
    [super dealloc];
}

- (void)loadView {
    [super loadView];
    [self updateLocationString:[map region]];
}

- (void)zoomIn:(id)sender {
    CLLocationCoordinate2D homeCoordinate = {37.783367,-122.402759};
    MKCoordinateRegion homeRegion = MKCoordinateRegionMakeWithDistance(homeCoordinate, 500, 500);
    [self.map setRegion:homeRegion animated:YES];
}

- (void)zoomOut:(id)sender {
    CLLocationCoordinate2D globeCoordinate = {30.145127,-40.078125};
    MKCoordinateRegion globeRegion = MKCoordinateRegionMakeWithDistance(globeCoordinate, 100000000, 100000000);
    [self.map setRegion:globeRegion animated:YES];
}

// Map View Delegate Method
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated {
    [self updateLocationString:[mapView region]];
}
@end
```

View
(Latitude, Longitude)
North America
Atlantic Ocean
Europe
Africa
South America
Google

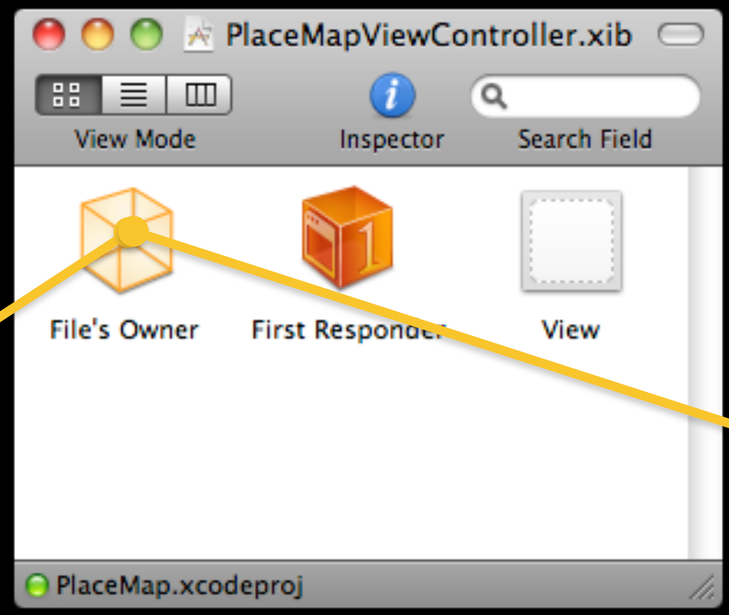
- (id)init {
if (self = [super init]) {
element = nil;
atomicElementView = nil;
atomicElementFlippedView = nil;
self.frontViewIsVisible=YES;
self.hidesBottomBarWhenPushed = YES;
}
return self;
}
- (void)loadView {
// create and store a container view
UIView *localContainerView = [[UIView alloc] initWithFrame:
[[UIScreen mainScreen] applicationFrame]];
self.containerView = localContainerView;
[localContainerView release];
containerView.backgroundColor = [UIColor blackColor];
}
.m

What object manages my views?

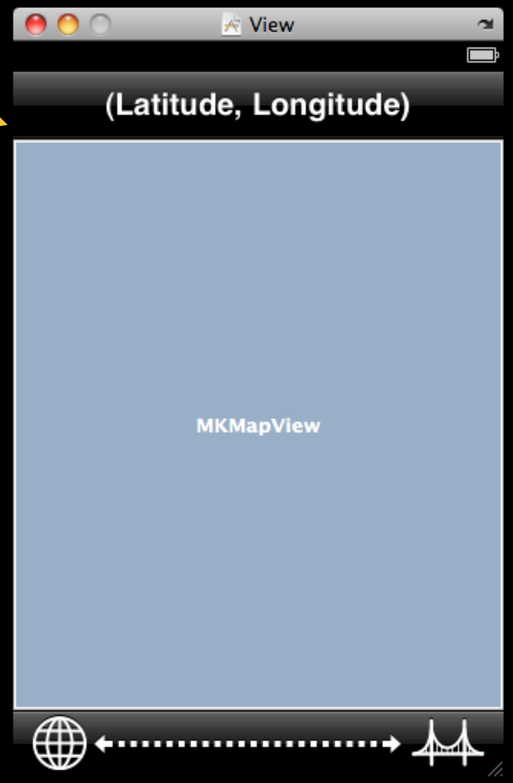
View Controller

How do I link my view controller to my view?

File's Owner

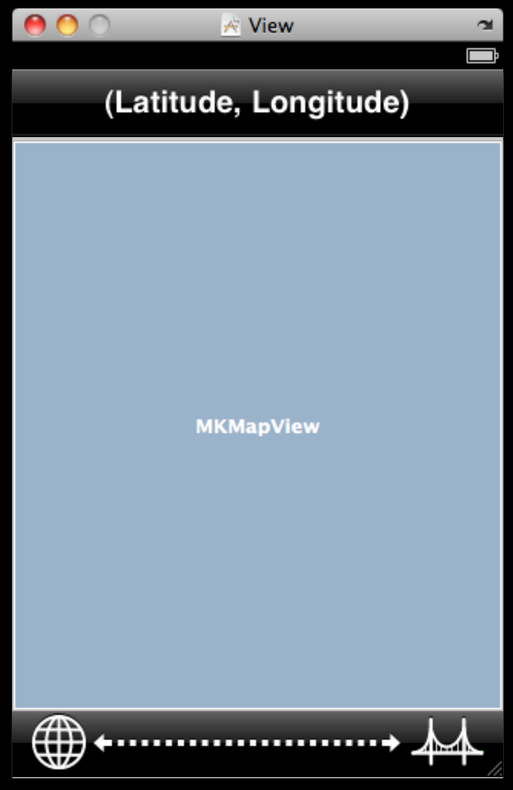

PlaceMapViewController.xib
View Mode
Inspector
Search Field
File's Owner
First Responder
View
PlaceMap.xcodeproj

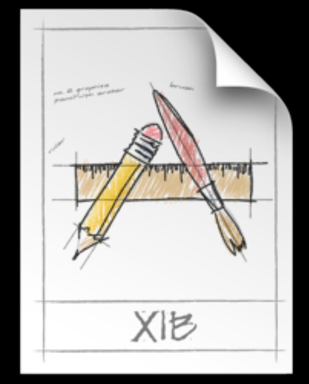
- (id)init {
if (self = [super init]) {
element = nil;
atomicElementView = nil;
atomicElementFlippedView = nil;
self.frontViewIsVisible=YES;
self.hidesBottomBarWhenPushed = YES;
}
return self;
}
- (void)loadView {
// create and store a container view
UIView *localContainerView = [[UIView alloc] initWithFrame:
[[UIScreen mainScreen] applicationFrame]];
self.containerView = localContainerView;
[localContainerView release];
containerView.backgroundColor = [UIColor blackColor];
}
.h


View
(Latitude, Longitude)
MKMapView

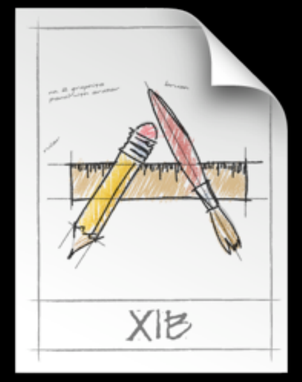
What happens when I save my interface file?

Saving Your Interface File

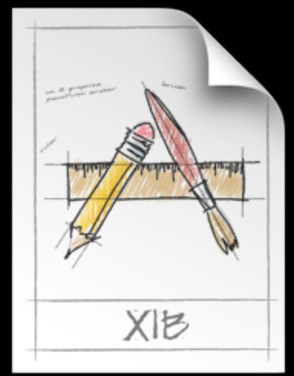




Saving Your Interface File



Saving Your Interface File



When does the view controller load my view?

Load View

You Respond to Delegate

```
- (NSArray *)loadNibNamed:(NSString *)name
[self.window addSubview:[self.mainViewController view]];
- (owner:(id)owner
options:(NSDictionary *)options;
```

Where can I configure my view?

View Did Load

View Controller

```
- (void)viewDidLoad {
    [super viewDidLoad];
    [self updateLocationString:[map region]];
}
```

What happens when my view is loaded?

Runtime

9:42 AM
(30.145127, -40.078125)
North America
Atlantic Ocean
Europe
Africa
South America
Google

View
(Latitude, Longitude)
North America
Atlantic Ocean
Europe
Africa
South America
Google

- (id)init {
if (self = [super init]) {
element = nil;
atomicElementView = nil;
atomicElementFlippedView = nil;
self.frontViewIsVisible=YES;
self.hidesBottomBarWhenPushed = YES;
}
return self;
}
- (void)loadView {
// create and store a container view
UIView *localContainerView = [[UIView alloc] initWithFrame:
[[UIScreen mainScreen] applicationFrame]];
self.containerView = localContainerView;
[localContainerView release];
containerView.backgroundColor = [UIColor blackColor];
}
.m

More Information

Michael Jurewitz
Developer Tools Evangelist
jurewitz@apple.com

Documentation
Mac OS X Human Interface Guidelines
http://developer.apple.com/ue

Interface Builder User Guide
http://developer.apple.com/documentation/DeveloperTools/Conceptual/IB_UserGuide/Introduction/Introduction.html

Related Sessions

iPhone Application Design Using Interface Builder	Pacific Heights Wednesday 10:30AM

Labs

Getting Started With Interface Builder	iPhone Lab A Wednesday 2:00PM
iPhone Application Design Using Interface Builder	iPhone Lab C Thursday 2:00PM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09